Mockito exercises

Exercise 1: Mocking and Stubbing

Scenario: You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

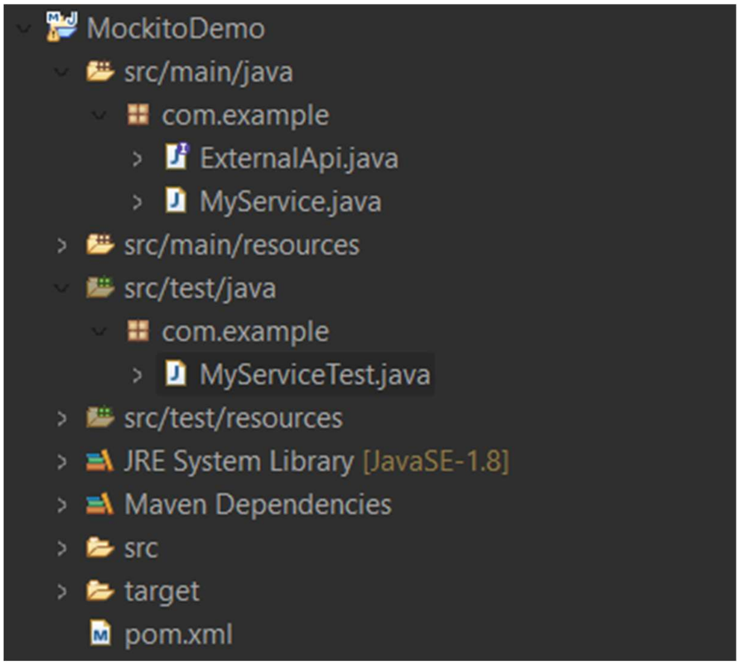
Steps:

1. Create a mock object for the external API.
2. Stub the methods to return predefined values.
3. Write a test case that uses the mock object.

Solution Code:

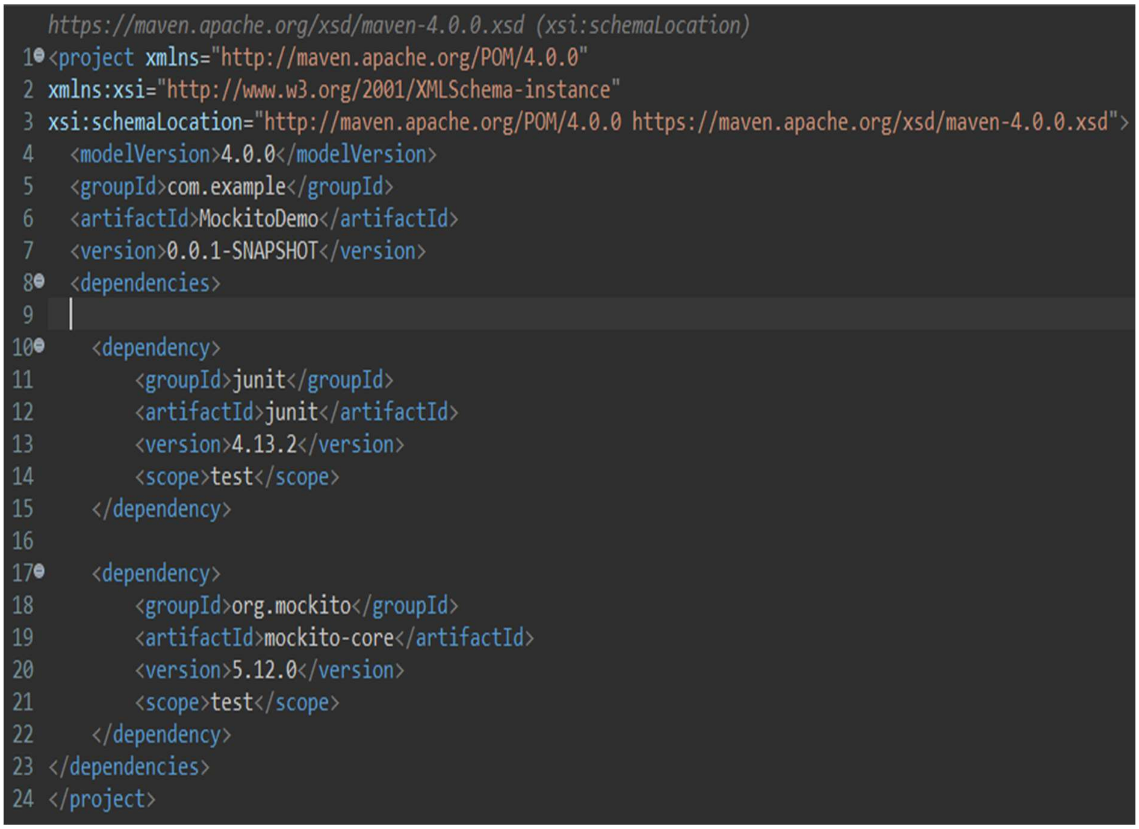
```
import static org.mockito.Mockito.*;

import org.junit.jupiter.api.Test; import org.mockito.Mockito;

public class MyServiceTest {

@Test public void testExternalApi() {

ExternalApi mockApi = Mockito.mock(ExternalApi.class);
when(mockApi.getData()).thenReturn("Mock Data");
MyService service = new MyService(mockApi);

String result = service.fetchData(); assertEquals("Mock Data",
result);

}

}
```

The below is the Project Explorer of my Maven Project



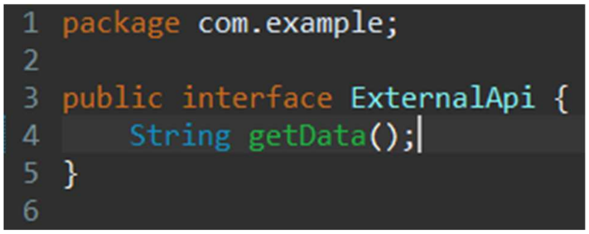
```
MockitoDemo
  src/main/java
    com.example
      ExternalApi.java
      MyService.java
  src/main/resources
  src/test/java
    com.example
      MyServiceTest.java
  src/test/resources
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  pom.xml
```

As I am using Maven project I have added dependencies to my pom.xml



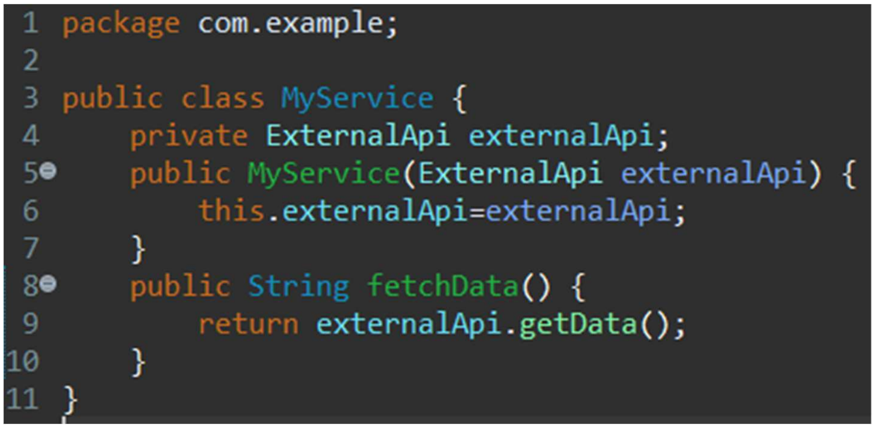
```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>MockitoDemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>

    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.2</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>5.12.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>
</project>
```

Created a class name ExternalApi.java



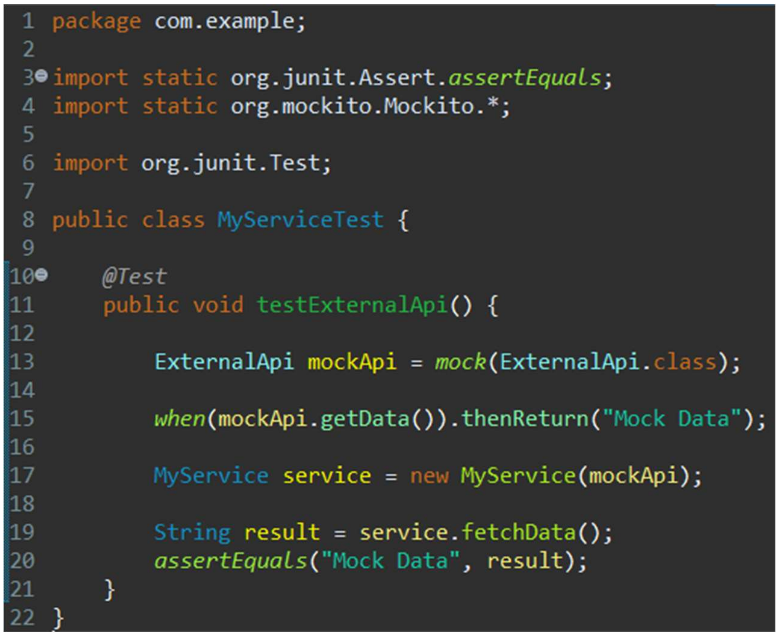
```java
package com.example;

public interface ExternalApi {
    String getData();
}

```

Created a class name MyService.java



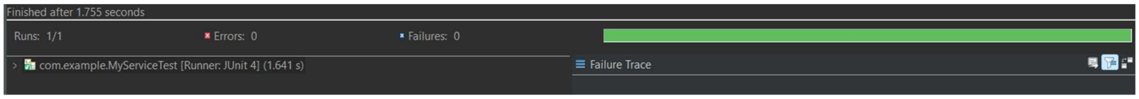
```java
package com.example;

public class MyService {
    private ExternalApi externalApi;
    public MyService(ExternalApi externalApi) {
        this.externalApi=externalApi;
    }
    public String fetchData() {
        return externalApi.getData();
    }
}
```

Created a class name MyServiceTest.java



```java
package com.example;

import static org.junit.Assert.assertEquals;
import static org.mockito.Mockito.*;

import org.junit.Test;

public class MyServiceTest {

    @Test
    public void testExternalApi() {

        ExternalApi mockApi = mock(ExternalApi.class);

        when(mockApi.getData()).thenReturn("Mock Data");

        MyService service = new MyService(mockApi);

        String result = service.fetchData();
        assertEquals("Mock Data", result);
    }
}
```

As the run was successfull for my test class I can see a greenbar with runs 1/1 and failures 0



Finished after 1.755 seconds

Runs: 1/1 Errors: 0 Failures: 0

com.example.MyServiceTest [Runner: JUnit 4] (1.641 s)